REMARKS

In this amendment no claims are added and claims 17 and 50 are cancelled. Accordingly, claims 1-16, 18, 34-49, and 51-59 are pending.

Claims 17 and 50 were rejected under 35 U.S.C. § 101 as being directed to non-statutory subject matter. These claims are herein cancelled thus obviating this rejection.

Claims 1, 34 and 59 were objected to as allegedly containing informalities. The Office Action asserted that the claim phrase "…temporally proximate to the operating being performed…" renders the claims indefinite. The Office Action then stated that above quoted phrase is being interpreted as "before the operation performed". Applicant respectfully traverses this objection. The present specification defines temporally proximate at ¶40, lines 3 and 4. Accordingly, the phrase is clear and definite and it is improper for the Office Action to interpret the phrase in a manner inconsistent with the specification.

REJECTIONS

(a) Claims 1-3, 9-12, 15 and 54-57 were rejected under 35 U.S.C. §103(a) as being unpatentable over Koshisaka, U.S. Patent No. 6,629,109 B1 (Koshisaka) in view of Dunphy, U.S. Patent No. 5,638,509 (hereinafter Dunphy) and further in view of Parasarathy, U.S. Patent No. 7,117,371 B1 (hereinafter Parasarathy).

(b) Claims 34-38, 43-51 and 57-59 were rejected under 35 U.S.C. §103(a) as being unpatentable over Dunphy further in view of Koshisaka and further in view of Parasarathy.

(c) Claims 39-42 were rejected under 35 U.S.C. §103(a) as being unpatentable over Dunphy in view of Koshisaka and further in view of Parasarathy and further in view of Midgely et al., U.S. Patent No. 5,608,865 (hereinafter Midgely).

Rejections (a) – (c)

A claimed invention is unpatentable for obviousness if the differences between it and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art. *In re Zurko,* 258 F.3d 1379, 1383 (Fed. Cir. 2001). Obviousness is a legal question based on underlying factual determinations including 1) the scope and content of the prior art, 2) the level of ordinary skill in the art, 3) the differences between the prior art and the claimed invention and 4) objective evidence of secondary considerations. Here the Official Action has erred in its factual assessment of points 1, 3 and 4 and, therefore, the rejections of all pending claims must be reversed. *Id at* 1386.

The Official Action made an incorrect factual finding that the claim term "operating system" was equivalent to the API of Koshisaka. The Official Action improperly relied upon this factual finding as a basis for its obviousness conclusions in all of the rejections. The Office Action fundamentally misconstrued the scope and meaning of the term API as used in Koshisaka. The record evidence, including Koshisaka, Dunphy, The Webster's New World Dictionary of Computer Terms[1], compel the conclusion that Koshisaka's API is not equivalent to the claimed operating system.

Koshisaka itself clearly distinguishes between operating systems and application programs. See elements 1 and 3 of Figure 1. More particularly, Koshisaka defines Application (from which API commands emanate) as generally used application software such as word processor software, spreadsheet software, CAD software, etc.

[1] Attached as Exhibit 1 to the Evidence Appendix of the Appeal Brief.

In contrast, Koshisaka defines operating system as software for operating the computer system, such as Windows 98.

Further in support of Applicants' position, one of the other cited references, Dunphy, discloses an operating system 19 that is separate and distinct from application programs 8. See Column 3, lines 36-47 and Figure 1.

Moreover, applicable technical dictionaries recognize the difference between APIs and operating systems. The Webster's New World Dictionary of Computer Terms defines "API" as a set of standards or conventions by which programs can call specific operating system or network services. The same dictionary defines the plain meaning of "Operating System" as a master control program that manages the computer's internal functions, such as accepting keyboard input, and that provides a means to control the computer's operations and file system. See pages 33 and 338 of Exhibit 1.

The Office Action alleges that Parasarathy discloses that its API is part of its operating system. For this assertion, the Office Action relies on Col. 5, lines 59-61. However, the cited passages of Parasarathy address neither operating systems nor APIs.

> ***"…hash is then encrypted using a private key that matches the public key located in the identity information component 17. The encrypted hash can be used to verify that the contents…"***

Col. 5, lines 59-61.

Nothing in this cited passage suggests to a person of ordinary skill in the art that an API is a part of the operating system.

Consequently, not a shred of evidence supports the Office Action's position that Koshisaka's API and the claimed operating system are the same. Because the Office

Action's conclusion of obviousness in each rejection was based on this erroneous assessment of the scope of the prior art, all of the rejections lack substantial evidence support and, for this reason alone, must be reversed. *Zurko* 1386.

Even if Parasarathy taught what the Office Action alleges, the Office Action must view the prior art as a whole through the prism of a person having ordinary skill in the art. Koshisaka, Dunphy, the computer dictionary, Mr. Williams Declaration and the present specification teach that the operating system is separate from the API. Accordingly, the prior art taken as a whole does not support and, in fact, teaches away from the Office Action's conclusions the API is part of the operating system.

The Rejection of Claims 1-3, 9-12, 15 and 54-57 under 35 U.S.C. §103(a)

Turning now to the individual rejections, the Office Action rejected claims 1-3, 9-12, 15 and 54-57 under 35 U.S.C. §103 as unpatentable over Koshisaka in view of Dunphy. This rejection is improper for the reasons set forth above as well as the following. The proposed combination of Koshisaka/Dunphy does not address all of the limitations of the claims. For the purpose of this discussion, claim 1 is representative of claims 2-3, 9-12, 15 and 54-57. Claim 1 is directed to a method for archiving files and recites "detecting an instruction by an operating system to perform an operation on an operating file". Notwithstanding the comments to the contrary in the Office Action, neither Koshisaka nor Dunphy, taken alone or in combination, teach or disclose this detecting step.

As mentioned above, the Office Action errs in equating Koshisaka's API with the operating system of the claims. The term "operating system" is defined in the instant specification at paragraph 28 as:

> **"A computer program that allocates system resources such as memory, disk space, and processor usage and makes it possible for the computer to boot up to a human user interface allowing the user to interact with the computer and control its operation".**

Where an explicit definition is provided by the applicant for a term, that definition will control interpretation of the term as it is used in the claim. *Toro Co. v. White Consolidated Industries Inc.*, 199 F.3d 1295, 1301, 53 USPQ2d 1065, 1069 (Fed. Cir. 1999). Here, this axiom is particularly applicable as the definition of the term "operating system" set forth in the specification parallels the plain meaning of the term as evidenced by Exhibit 1 attached to the appeal brief and the understanding of the skilled artisan as evidenced by Koshisaka and Dunphy.

Koshisaka, teaches "an application-centric" file revision management system and method by which file revision management allegedly can be implemented even if applications are not provided with file revision management functions. Koshisaka teaches the use of an Applications Programming Interface (API) layer to be placed between an application and an operating system. According to Koshisaka, file backup reliability of the applications can be improved by this file revision management system. *See* Koshisaka, column 1, lines 57-63. *See also* Exhibit 4, Paragraph 5. A file manipulation monitoring section in Koshisaka first detects file manipulation (e.g., file deletion or file name change) that is to be executed by the application by intercepting these instructions as they are generated by the application. *See* Exhibit 4, Paragraph 6. Koshisaka specifically states, "the file manipulation monitoring section constantly monitors API (Application Program Interface) commands which are outputted by the application 1 to the operating system and thereby detects the file manipulation which is (going to be) executed by the application 1." *See* Koshisaka, column 6, lines 34-38.

The method of the present invention, as defined by claims 1 and 54, is a "data-centric" approach to file archiving, not an application-centric approach. This distinction is evidenced by the claim limitations of, "detecting an instruction ***by an operating system*** to perform an operation on an operating file," and "capturing the operating file temporally proximate to the operation being performed on the operating file, responsive to the detection of the instruction."

In direct contrast, Koshisaka is not data-centric; that is, it does not teach detection of an instruction by an operating system. Rather, the detected ***instruction*** or command in Koshisaka is ***the API command*** that is generated ***by the application***, not ***the operating system***. *See* Exhibit 4, Paragraphs 5, 6 and 7. For example, in Koshisaka, when an API command requesting file deletion is output by the application, the command is detected and hooked by the file manipulation monitoring section in the file management system 2. Subsequently, the processing section sends a different API command to the operating system. In other words, the instruction is first detected by the API and hooked. After the instruction is hooked, a different instruction is passed to the operating system, and the original command sent by the API to the operating system is not executed during performance of Koshisaka's revision management system activities. *See* Williams declaration, Paragraph 5.

As a result of detecting the instruction ***by the application***, unlike the present invention, it is believed that Koshisaka provides a very limited layer of file protection, as file protection occurs only if the application is compatible with Koshisaka's assumptions of API behavior. See, Koshisaka, column 7, lines 59-64. *See also* Exhibit 4, Paragraph 8. However, because the invention as defined by claim 1 is captures files based on operating system instructions, it achieves file protection of intended file change operations as well as file changes that result from some type of file corruption.

Dunphy is directed to a data storage and protection device used for data backup. Dunphy, like Koshisaka, teaches an application-centric solution. As illustrated in Figure 1, Dunphy depicts an operating system 19, application programs 8 and file system 9. Data file monitor 11 is interposed between application programs 8 and file system 9. Data file monitor 11 intercepts communications between ***application programs 8 and file system 9***. Data file monitor 11 reviews the communication to determine whether it relates to a data file that the user has selected for monitoring. If the data file is one to be monitored, it is determined whether the communication results in a change in content of a data file. If data file change is detected, data file monitor 11 extracts data file status and activity information from the received communications and uses this data to build an event log 12. Data file monitor 11 also determines whether the communication requests an operation that changes the contents of the data file, i.e., would cause a loss of data. If so, then the data file is saved. *See* Dunphy, Column 3 line 49 to column 4, line 21.

Unlike the method of claim 1, neither Dunphy nor Parasarathy teach detecting an instruction ***by an operating system.*** The portion of Parasarathy that the Office Action relies on for such a teaching does not even mention APIs or operating systems. Thus, Dunphy and Parasarathy suffer from the same deficiencies as Koshisaka.

Since the combination of Koshisaka, Dunphy and Parasarathy does not address all claim limitations, the rejection of claims 1-3, 9-12, 15 and 54-57 must be reversed.

The Rejection of claims 34-38, 43-51 and 57-59

The Office Action rejected claims 34-38, 43-51 and 57-59 under 35 U.S.C. §103 as unpatentable over Dunphy in view of Koshisaka and further in view of Parasarathy. This rejection is improper for the reasons set forth above as well as the following. The proposed combination of Dunphy, Koshisaka and Parasarathy does not address all of the limitations of the rejected claims.

Claim 34 is directed to a method for archiving files including the steps of detecting an instruction by an operating system to perform an operation on an operating file and creating an archive file from the operating file and storing the archive file in a temporary storage location temporally proximate to the operation being performed on the operating file and responsive to detecting the instruction. As mentioned above in the discussion of the rejection of claims 1-3, 9-12, 15 and 54-57, neither Dunphy, Parasarathy nor Koshisaka, taken alone or in combination teach or suggest detection of an instruction ***by an operating system.*** In addition, neither Dunphy, Parasarathy nor Koshisaka teach storing an archive file created from the operating file in a temporary first storage location responsive to detection of the operating system instruction. Koshisaka teaches that, upon detection of only one of a delete command or a file rename command from the ***application***, the deleted file name is stored and a corresponding back up file name is stored in memory. See Koshisaka, column 7, line 52 to column 9, line 43.

Unlike the method of claim 34, Dunphy **does not** detect an instruction ***by an operating system***. Dunphy, much like Koshisaka and other references of record, is concerned with communications from the application programs themselves. As explained in detail in connection with the discussion of Koshisaka above, the present invention as defined by claim 34 performs file capture and file manipulation based on instructions from the operating system. This is a significant advance because it allows file capture before the file operation is performed, for example. Dunphy is limited to addressing files for which an operation that changes the file content is specified, e.g., a delete or modify operation. Dunphy would be useless against operations that do not intentionally change file content but that may corrupt file content such as file open operations or file rename operations.

The cited passages of Parasarathy have no apparent relevance to the claims.

It is readily apparent that the Office Action erred in its factual findings of the differences between the Dunphy/Koshisaka combination and the subject matter of claim 34. In view of this error, the Office Action failed to establish a *prima facie* case of obviousness as to claims 34-38, 43 and 59.

i. Claim 44

With respect to claim 44, it requires that the archive file pass through two storage locations before ending up in permanent storage (its third storage location). The Office Action cites to column 4, lines 25-67 of Dunphy as teaching the method of claim 44. However, the cited portion of Dunphy does not provide such a teaching.

Lines 24-38 of Dunphy discuss creation of an event log 12. Event log 12 is not an archive file. Rather it is a collection of data that includes identifying information about a file. The closest thing that Dunphy teaches to an archive file is the data file saved in stash can 13. However, Dunphy does not teach or suggest moving that data file from stash can 13 to an intermediate storage location and subsequently to a permanent storage location as required by claim 44. Koshisaka provides no additional teaching that would have rendered the subject matter of claim 44 obvious to the skilled artisan.

The Office Action's explanation of the operation of Dunphy on page 5 is unsupported by Dunphy itself. Database 14 and event log 12 are part of protection system 10. Accordingly, the explanation provided by the Office Action on page 5 is incorrect.

It is apparent that the Office Action erred in its factual findings of the differences between the Dunphy/Koshisaka/Parasarathy combination and the subject matter of claim 44. In view of this error, the Office Action failed to establish a *prima facie* case of obviousness as to claims 44-51 and 58.

The Rejection of Claims 39-42

The Office Action rejected claims 39-42 under 35 U.S.C. § 103 as unpatentable over Dunphy in view of Koshisaka and further in view Parasarathy and further in view of Midgely et al., U.S. Patent No. 5,608,865 (hereinafter Midgely). This rejection is improper for the reasons set forth above as well as the following reasons. The proposed combination of Dunphy, Koshisaka, Parasarathy and Midgely, does not teach all of the limitations of claims 39-42.

ii. Claims 39 and 40

Claim 39 calls for searching a first storage location for the archive file responsive to receipt of a message from a timer. Accordingly, the storage location is searched at some specified time interval. The Office Action recognizes that neither Koshisaka nor Dunphy teach searching a storage location for the archive file responsive to a message from a timer. The Office Action asserts that Midgely suggests notifying a user that a file change is about to be made via a message from a timer. Office Action, Page 18. Applicants submit that the Office Action's assertion about Midgely's teachings are completely unsupported. Nowhere does Midgely even remotely indicate to a person having ordinary skill in the art that a temporary file location is searched responsive to a message from a timer as required by claim 39. The Office Action references a passage of Midgely, specifically column 7, lines 59-63, but this passage has nothing to do with the relevant claim limitations. At best, Midgely teaches generating a notification when a user when a user requests a file open operation. It does not suggest the claimed operation of searching a temporary storage location responsive to any type of notification, whether that notification is a message from a timer as in claim 39 or a message from a resident program as in claim 40. Accordingly, the Office Actions failed to establish a *prima facie* case of obviousness for claims 39 and 40 and the rejection of claims 39 and 40 must be reversed.

iii. Claims 41 and 42

Regarding claims 41, it calls for moving the archive file to a permanent storage location responsive to a message from a timer. Claim 42 calls for moving the archive file to a second storage location responsive to a message indicating when the second storage location is available. As recognized by the Office Action, neither Koshisaka nor Dunphy teach a method of archiving a file wherein archive files are moved to a second storage location responsive to receipt of a message. While the Office Action relied on Midgely for teaching that an agent is notified when a client requests a file open operation, prior to executing the open operation, Midgely offers no teaching that is remotely related to the limitations of claims 41 and 42. More particularly, Midgely does not teach moving an archive file responsive to a permanent storage location responsive to a timer message or a message indicating availability of the permanent storage location. Accordingly, the Office Action failed to establish a *prima facie* case of obviousness for claims 41 and 42 and the rejection of claims 41 and 42 must be reversed.

Conclusion

In view of the foregoing amendments and remarks, it is respectfully submitted that the pending claims are allowable. Reconsideration of the rejections and a favorable action on the merits are respectfully requested.

Respectfully submitted,

CAHN & SAMUELS, LLP

By: /Frederick Samuels/
Frederick N. Samuels, Esq.
Reg. No. 34,715
1100 17th Street, N.W., Suite 401
Washington, D.C. 20036
(202) 331-8777 (Telephone)
(202) 331-3838 (Facsimile)
Attorney for Applicants